ADG Best Practices – Programming

C# Programming

- When using enums, break them into separate files unless they should be logically grouped into a single file.

Entity Framework

- The Code-First approach should be utilized for all new projects without an existing database, however the Database-First approach may be appropriate in circumstances where a database already exists.
- Auto-migrations should be disabled by default.
- Create an interface to be implemented by the DbContext so to enhance testability and so that it can be injected by Dependency Injection into services.
- Each database table should be represented in the DbContext as a DbSet.
- An interface should be used to reflect all DbSets that are in the context file.
- Create an initializer that inherits from CreateDatabaseIfNotExists<DbContext> to seed the database with initial data.
- When defining relationships between entities, DataAnnonations are the preferred method, but when necessary it's acceptable to use the FluentAPI as well.
- Foreign Keys should be represented with an integer property, followed by a virtual navigation property of the model to enable LINQ expressions.

MVC Controllers

- Actions should start with an explicit attribute, such as [HttpGet] or [HttpPost].
- Controllers should utilize ViewModels for sending data to Views, and use ViewBag sparingly, while avoiding ViewData unless it's necessary.
- Business logic should be located in a separate service layer if possible and avoid using business logic in controllers.

Other MVC Functionality

- When assets are not coming via the ADG CDN, bundling should be utilized in the BundleConfig.cs.